

Please replace the paragraph beginning at page 9, line 20, with the following rewritten paragraph:

AB
-- Before starting the negotiation, a pre-negotiation stage may be necessary (500). The product or service package under negotiation has a set of issues that make part of a common ontology. The pre-negotiation stage may establish which issues be part of any offer and which are optional. Other preparations or updates may be the object of this stage too. For instance, in this stage, previously established behavior, constraints, intervals of acceptability for offers, etc. may be updated in order to take into account the particular partner in negotiation or the overall strategy in negotiation. In one embodiment of the present invention, a user may, via a user interface (through, for example, a browser) configure a buyer agent for negotiation. The user may be required to select the package the user desires to buy from a list of packages each one containing goods or services. Once the particular package is chosen, the negotiable terms or components of the negotiation are established. For example, if the user wishes to buy a particular make and model of a car, the price, term of warranty, color, options, etc. may be negotiable. If the user wishes to subscribe to a wireless telephone service, price per minute, monthly connect charges, and the number of free minutes, etc. may be negotiable. --

Please replace the paragraph beginning at page 12, line 28, with the following rewritten paragraph:

A4
-- The next element to consider is the relative importance that an agent assigns to each issue (sub-object or attribute) under negotiation. W is the importance of issue j for agent i . The weight associated with each issue is defined for each agent at the beginning of the negotiation, but can be changed later on in the negotiation (by external causes, the agent's interests, human intervention, inference based on a set of business rules, specified transition functions for the weights or for the overall behavior, etc.). Considering the negotiation package as a tree with sub-objects represented as non-terminal nodes and attributes of the parent sub-objects represented as terminal nodes, we assume that the weights related to the decomposition of a particular sub-object in a tree for both agents are normalized at any moment t i.e.

(2) $w_j^i(t) = 1$, for $i \in \{a, b\}$, $j = 1, \dots, n$ being the component of a particular sub-object in a tree. --

Please replace the paragraphs beginning at page 13 line 15, with the following rewritten paragraph:

A5
-- $w_j^i(t)s_j^i(x_j)$, where $w_j^i(t)$ is the importance of the component j , x_j is its value (obtained from the scorings of its sub-components) of a particular sub-object for agent i at time t . The scoring of an attribute is obtained by taking into account its value and its assigned scoring function. --

Please replace the paragraph beginning at page 14, line 9, with the following rewritten paragraph:

A6
-- Regardless of the mechanism implemented according to the example embodiment of the present invention, the agent uses the same method to decide to accept an offer or to propose a new counter-offer. Referring again to fig. 5, when the agent receives an offer (570), the agent evaluates it (575), and then generates the counter-offer it intends to propose in accordance with the particular mechanism that it has implemented (580). The agent then evaluates the generated counter-offer. If the score of the received offer is greater or equal than the score of the offer it intend to propose (585, 586)), the agent may accept the offer and send an ACCEPT message to the opponent agent (590), thus successfully completing the negotiation thread (560). Otherwise, the agent proposes the new generated counter-offer (587, 595). When an agent is interested in obtaining only one contract agreement, the agent sends an EXIT message to all the other negotiation partners after it successfully terminates a negotiation thread (565). Otherwise, the agent continues to negotiate in the other threads until the established deadline or until another agent event causing the end of negotiation, for example user intervention, no negotiation partners left in the market, etc. This would be the case for a seller, for example, whose concrete strategy and the number of agreements it intends to obtain depends also on the quantity of goods or services it has in stock.--

Please replace the paragraph beginning at page 16, line 25, with the following rewritten paragraph:

A7
-- **Example Negotiation System:** In accordance with one embodiment of the present invention, the negotiation model described above is integrated in a scalable software platform, such as, for example, an Internet negotiation service for different subscribers interested in negotiations using intelligent agents. These subscribers may be individual customers or business customers that intend to sell or buy goods or services. The interconnectivity capabilities offered also allow other applications to connect to the negotiation platform.--

Please replace the paragraph beginning at page 18, line 3, with the following rewritten paragraph:

A8
-- The platform 650 receives inputs from users (e.g., subscribers 600-602 and administrators 605, 610) via a user interface 705. In the case of business subscribers 710, input may also be received from specific applications or information systems. Specific outputs include the results of negotiations, database logs, messages sent to users (for example via e-mail, SMS, WAP, etc.). --

Please replace the paragraph beginning at page 18, line 9, with the following rewritten paragraph:

A 9

-- In accordance with the example embodiment of the present invention, the interaction with the user is handled by the User Interface module (705), which uses the Application Program Interface of the web server (Netscape Server Application Program Interface (NSAPI) or Internet Server Application Program Interface (ISAPI), for example). The User Interface module gets the user's request, manipulates temporary data involved in HTTP or WAP request (which specify the actions to be executed and any associated parameters) and generates the appropriate XML streams or documents. These documents contain the information to be presented to the user (XML defines a way of structuring the information, without involving the presentation). In order to render the information in an appropriate form on the user device (computer, WAP phone, etc.), the XML documents are sent to an extensible style language (XSL) processor represented by the XSLProcessor module 706, together with the corresponding XSL files that were created for presentation purposes. The XSLProcessor outputs for example HTML pages (if the user device is an computer) or WML pages (if the user uses a WAP phone), which are sent back for presentation in a specific browser on the user device through the User Interface module.--

Please replace the sentence beginning at page 18, line 17, with the following rewritten sentence:

A 10

--The XSL processor XML streams or documents from the NSAPL(ISAPI) modules and outputs HTML pages or WAP pages based on the appropriate XSL files. --

A 11

Please replace the paragraph beginning at page 20, line 26 and ending at page 21, line 5, with the following rewritten paragraph:

-- The Load Balancer module 820 provides better performance for the entire negotiation environment, by distributing the tasks on different physical machines, using a strategy that allows smooth control of the network and processor resources involved. Load Balancer 820 receives requests from the user interface module 705 to start new negotiation processes. If agents involved in negotiations do not yet exist, the Load Balancer 820 asks one of the Agent Factories 825 to create them, using the behaviors dictated by their owners and/or set by rules. To improve performance, the Load Balancer 820 may decide to move an agent from one machine to another. For example, if all the other agents negotiating with a particular agent are running on a different machine, moving that agent to the other machine reduces network overhead. The Load Balancer 820 may run on a different physical machine than those of the Agent Factories 825 and Marketplaces 835. The communication between the components is performed transparently using, for example, CORBA technology. --

Please replace the paragraph beginning at page 21, line 26 and ending at page 22, line 3, with the following rewritten paragraph:

A 12

-- The Marketplace component 835 provides the environment in which the agents meet each other. This component supervises the communication between the parts that

participate in negotiation. It acts as a mediator in negotiations between parties. The Marketplace 835 is also responsible for notifying the users, the agent owners, and the system about events that occur during the negotiation process. The messaging module 715 sends the actual notifications. In accordance with the example embodiment, the seller and buyer agents communicate with each other indirectly by means of the Marketplace 835 if they do not run on the same machine, CORBA calls are used.--

Please replace the sentence beginning at page 22, line 5, with the following sentence:

A13
-- The Marketplace 835 collects data on the progress of negotiations, and send it to the Database component 720. Some of this data may be retrieved later for profiling and statistics.--

Please replace the paragraph beginning at page 24, line 5, with the following rewritten paragraph:

A14
-- The user may configure his/her own agent using WAP (for a basic configuration) or HTTP (for a more advanced agent configuration). The configuration is sent to the User Interface which stores it in the Database. For business customers, authorized administrators may use the same interface, or may set up business rules using another interface.--

Please replace the sentence beginning at page 24, line 18 with the following rewritten sentence:

A15
--1. To start a negotiation 1005, a subscriber 1000 sends a request (via WAP or via HTTP) to the User Interface, specifying the agent to act on his/her behalf. The User Interface contacts the Negotiation Engine via the NegoEngine Access component 810.--

Please delete the paragraphs beginning at page 25, lines 9-14 and page 25, lines 16-20.

Please replace the paragraphs beginning at page 25, lines 21-30 with the following paragraph.

A16
-- **Request Negotiation Thread Status:** Fig. 12 shows the sequence of interactions and the actors or main components involved when online information about a negotiation thread is requested. Online information about a negotiation thread where a specific agent (acting on behalf of a specific user for example) is involved, may be obtained through the Online Negotiation Information Server 815, which can be contacted via TCP/IP by an applet 1205 running on the client side (in a HTML browser for example) (1210, 1215). The Online Negotiation Information Server 815 contacts the Load Balancer 820 (via CORBA) and requests the status of the specified negotiation thread. The Load Balancer physically identifies (using its CORBA IOR—Interoperable Object Reference) the agent and asks it about the status of the specified negotiation thread. The agent's response is sent back on the same path till reaches the applet originating the request. By using such a procedure, for example, a user may obtain dynamic information in a graphical form related to the negotiation thread(s) in which one of his or her agents is involved.--

Please replace the table on page 27 with the table below.

Action Type	Meaning	Sent by	Sent when
Ask	A buyer asks for the object described in the packet (the packet contains an offer or counteroffer)	Buyer	The current offer was not accepted but the agent wishes to continue the negotiation
Offer	A seller wants to offer the object described in the packet (the packet contains new offer)	Seller	The last buyer Ask was not accepted but the seller wishes to continue the negotiation
Accept	An agent wants to accept the object proposed by its partner (the packet contains the offer)	Buyer or Seller	The utility of the object received is bigger than the utility desired at current moment
Confirm	An agent confirms that the accepted offer hasn't expired.	Buyer or Seller	An agent received an Accept message and still wants to buy/sell the product in the terms specified in previous step of then negotiation thread
Reject	An agent cannot confirm the offer or ask sent in the previous negotiation step.	Buyer or Seller	An agent received an Accept message but its internal policy has changed, so it will not confirm the previous request, but still wants to continue the negotiation thread
Exit	An agent decided to terminate the negotiation with the current partner	Buyer or Seller	The deadline expired, the buyer already bought the product from another seller, or the seller's stock is zero and the seller cannot continue negotiating or the agent has been stopped by the owner.
Syntax Error	An agent didn't understand the object description	Buyer or Seller	The ontology is not recognized by the agent
Semantic Error	An agent didn't follow the negotiation protocol (e.g., sent to Confirm after an Ask)	Marketplace	The marketplace detects an inconsistency in the protocol